Object Detection Using Correlation Technique

[1]Ramesh Uttam Kadam, [2]Amit Singh, [3]Dhwani Chitalia, [4]Pragati Upadhyay
[1]Student, [2]Student, [3]Student, [4]Assistant Professor
[1]Electronics Engineering,
[1]Shree L R Tiwari College of Engineering of engineering, thane, India

***Abstract* - This paper describes an object detection using correlation and template matching technique. This paper contains how we can use simple correlation technique for detection of object by using matlab software and image processing. This includes technique about cropping image, template making from image, various keywords useful for correlation programs and explanation about them.**

Object detection in computer vision is the task of finding a given object in an image or video sequence. The aim of object detection is to correctly identify objects in a scene and estimate their pose (location and orientation). The goal is to realize the ability of current object detection techniques to find similar objects when input is entirely in image form. In this present work, template matching techniques is used to recognize the object using correlation. Here we mention each step which required detecting an object by using correlation technique with explanation, examples and outputs carried out on matlab software.

***Index Terms*—Detection, Correlation, Normalized cross correlation, Template Matching**

I. Introduction

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Human recognize large amount of object in images very easily, regardless of the fact of different viewpoint of object as like in size, scale or they are in translated or in rotated [1]. But for computer vision this detection of different viewpoints are very challenging. So we are using correlation and template matching method for detecting of different object viewpoints as like example ,Any particular face detection or human detection, any object like bottle, key, watch detection from images, particular vegetable detection. The object detection technique is very good for real life application if we want to detect any wanted face detection or object detection through CCTV camera we don't need to watch all video or image sequence. Just run program if face is there it will detect automatically.
Object detection can be go through by following process:

1. At staring process for detections trained on sample images of the target object class and other objects.
2. Differentiate the target object class and other objects.
3. When new images are fed the system can sense the presence of the target object class.

The object detection problem can be separated into two basic blocks:

- low level vision
- High level vision.

The low level vision: It can be seen as to isolate objects and regions from the given image and similarly extracting other characteristic features from an image.
The high level vision: It means the interpretation of required objects or features in the frame of a reference scene.
For low level vision we have to create template image which is representing a similar image of object which we have to extract from newly fed image in the system. Then we have to match that template with new image using template matching.

Template Matching: It is a Technique used to categorize objects. A template is a small image (sub-image) .The goal is to find occurrences of this template in a larger image that is, you want to find matches of this template in the image. Template matching techniques compare portions of images against one another. Sample image may be used to recognize similar objects in source image.

Template matching has been a classical approach to the problems of locating and recognizing of an object in an image. Template matching technique, especially in two dimensional cases, has many applications in object tracking, image compression, stereo correspondence, and other computer vision applications [6, 7, 8, and 9]. Even now, it is a fundamental technique to solve them.

Among several matching methods, Normalized Cross Correlation (NCC) and square root of Sum of Square Differences ~SSD) have been used as the measure for similarity. Moreover, many other template matching techniques [6,7,8] , such as Sum of Absolute Differences (SAD) and Sequential Similarity Detection Algorithm (SSDA) have been adopted in many applications for pattern detection, video compression and so on. In addition, template matching has been extensively used in various applications, for example, extraction of container identity codes [10], image segmentation, [11] and so on.

To compare the both images and finding similarity between both images we are using correlation technique.Correlation is a measure of the degree to which two variables, not necessary in actual value but in general behavior. The two variables are the corresponding pixel values in two images, template and source.

II. METHODOLOGY

Normalized cross correlation

Normalized cross correlation is a well-liked method for finding 2D patterns in images. An (2*h* +1) × (2*w*+1) template **T** is correlated in opposition to an image **X**. At the image location (*u*, *v*), the normalized cross correlation is computed as

$$\mathbf{c(u,v)} = \frac{\sum_{i=-h}^{h}\sum_{j=-w}^{w} (\mathbf{X(i,j)} * \mathbf{T(i,j)})}{\sqrt{\sum_{i=-h}^{h}\sum_{j=-w}^{w}\mathbf{X(i,j)}^2}\ \sqrt{\sum_{i=-h}^{h}\sum_{=-w}^{w}\mathbf{T(i,j)}^2}} \tag{1}$$

With

$$\mathbf{X(i,j) = x(u+i, v+j) - \bar{x}}$$

$$\mathbf{T(i,j) = t(u+i, v+j) - \bar{t}}$$

In [12], a fast algorithm was developed to compute the denominator term as

$$\sum_{i=-h}^{h}\sum_{j=-w}^{w} \mathbf{(x(u+i, v+j) - \bar{x})^2} \tag{2}$$

This is achieved by observing that this term can be decomposed into three parts:

$$\sum_{i=-h}^{h}\sum_{j=-w}^{w} \mathbf{x(u+i, v+j)^2}$$

And

$$\bar{\mathbf{x}}\sum_{i=-h}^{h}\sum_{j=-w}^{w} \mathbf{x(u+i, v+j)}$$

And

$$\mathbf{(2h+1)(2wv+1)\bar{x}} \tag{3}$$

The first two terms can be computed efficiently using integral images of the original image and the squared image. To speed up the numerator computation, we can decomposed the template into box basis function so that,

$$\mathbf{t \approx \sum_{i\in\Lambda} \alpha i \phi i} \tag{4}$$

Then the numerator becomes,

$$\boldsymbol{X(u,v)t = \sum_{i\epsilon\Lambda} \alpha i(\phi i x(u,v))} \tag{5}$$

This can be computed using | Λ | multiplications and4 | Λ | −1 additions [2].

III. Steps for object detection:

i. Template making.

Use ‘imcrop(image)’ command for cropping template from main image or use ‘imtool(image)’ command and crop image by selecting required area and save image as template.

Step 1: Read main image using ‘imread(Image)’ command.

‘imread (main image)’

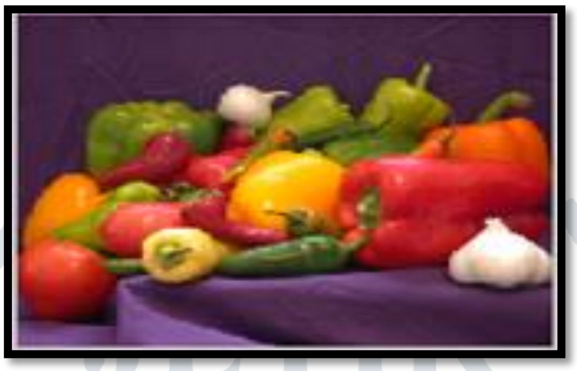


Figure 1: Main image

Step 2: Crop the part from the main image for making template.

Use direct command ‘imcrop(image)’ or use ‘imtool (Image)’ command ang select object which want to crop.

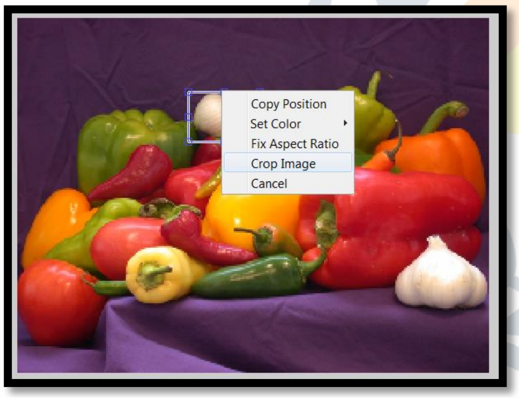


Figure 2: Main image

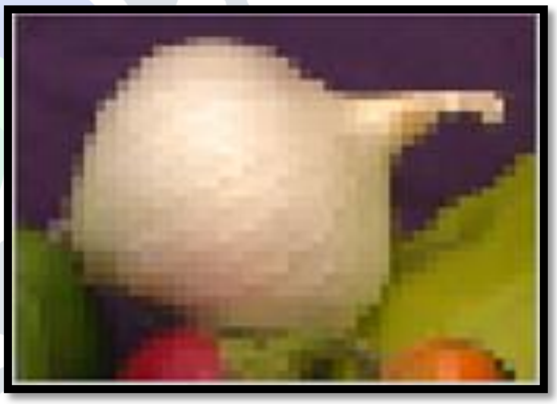


Figure 3: Cropped image

Step 3: Save the cropped image as template for detecting same object from main image.

Figure 2 : Crop Image (template image)

ii. Correlation

Step 1: Choose Sub regions of Each Image (template image as well as main image)

It is important to choose sub regions which are similar of both the images. The template's sub-image must be smaller than the original's sub image. We will get these sub regions using either the non-interactive script **or** the interactive script.

Interactively script:
This interact the program in between the process.
Example:
[sub_template, rect_template] = imcrop(template);
[sub_main, rect_main] = imcrop (main);

Non interactive script:
Save the rectangular position of that crop image and main image in a program. This position is use at the time of recovering matched image from main image.
Example:
rect_template = [0.5 0.5 46 34];
rect_main = [0.5 1.5 286 214];
sub_template = imcrop(template,rect_template);
sub_main = imcrop(peppers,rect_main);

. We can get this position at the time of cropping image by right clicking on top middle portion of rectangle on image.

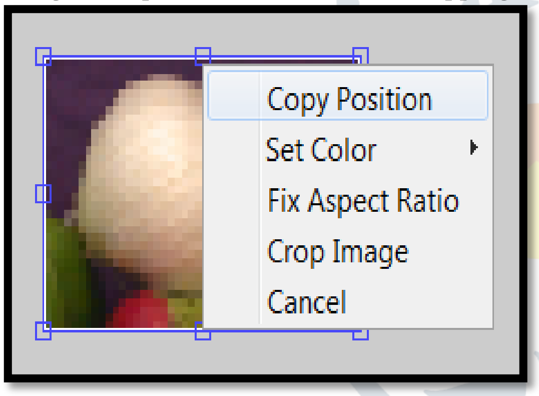


Figure 3 : Sub-image of template

rect_onion = [0.5 0.5 46 34];

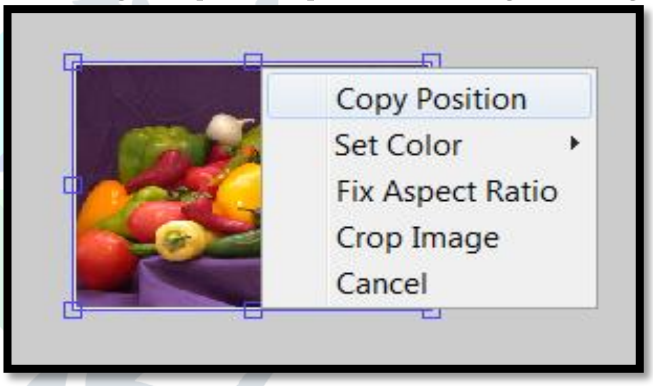


Figure 4 : Sub-image of main image

rect_peppers = [0.5 1.5 286 214];

Step 2: Calculate the normalized cross-correlation of two sub images and display it as a surface plot.

We are using this function for finding highest matched peak value from both sub images. The peak of the cross-correlation matrix occurs where the sub images are best correlated.

Normxcorr2 only works on grayscale images, so we pass it the red plane of each sub image by fallowing command.
As example,

'sub_template (:,:,1), sub_main(:,:,1)'

C = normxcorr2(template, A) :Computes the normalized cross-correlation of the matrices template and A. The matrix A must be larger than the matrix template for the normalization to be meaningful. The values of template cannot all be the same. The resulting matrix C contains the correlation coefficients, which can range in value from -1.0 to 1.0.
If images are similar then it gives highest peak value as 1.0.

Use of surf function and shading function is optional we can detect object without using these two functions but for understanding correlation function graphically we are giving in this paper.

Use **surf and surfc:** To view mathematical functions over a rectangular region.
surf and surfc create colored parametric surfaces specified by X, Y, and Z, with color specified by c or z.

surf(Z) creates a three-dimensional shaded surface from the *z* components in matrix Z, using x = 1:n and y = 1:m, where [m,n] = size(Z). The height, Z, is a single-valued function defined over a geometrically rectangular grid. Z specifies the color data as well as surface height, so color is proportional to surface height.

shading flat: The shading function controls the color shading of surface and patch graphics objects .Each mesh line segment and face has a constant color determined by the color value at the endpoint of the segment or the corner of the face that has the smallest index or indices.

Example:

```
sphere(16)
axis square
shading flat
title('Flat Shading')
```

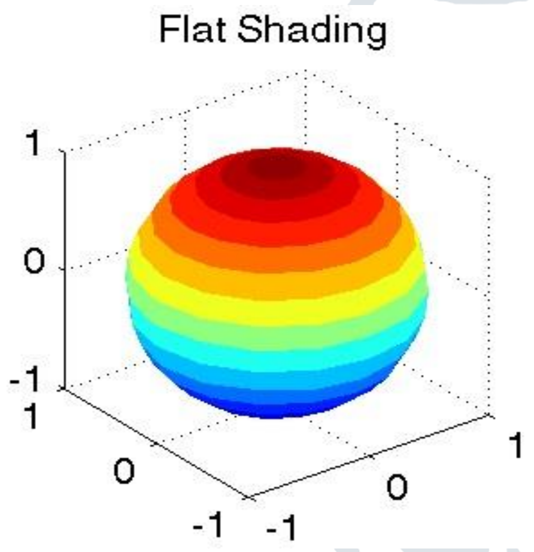


Figure 5 flat shading

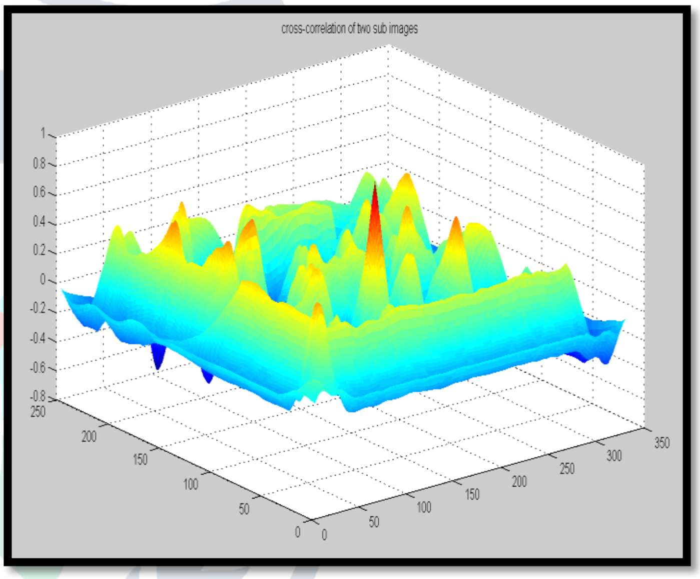


Figure 6 : correlation view in 3D

Step 3: Find the Total Offset between the Images

The total offset or translation between images depends on the location of the peak in the cross-correlation matrix, and on the size and position of the sub images.

To find total offset we have to find correlation offset and second one is relative offset.

A)For finding correlation offset following steps will be followed:

[C,I] = max(A) :this gives the maximum values of A, and returns them in output vector I. If there are several identical maximum values, the index of the first one found is returned.
Example: Two similar images gives-
max_c = 1.0000…maximum peak value and imax = 488919….. Identical values returned index

[I,J] = ind2sub(siz,IND): [ypeak, xpeak] = ind2sub(size(c),imax(1))

Returns the matrices I and J containing the equivalent row and column subscripts corresponding to each linear index in the matrix IND for a matrix of size siz. Siz is a vector with ndim(A) elements (in this case, 2), where siz(1) is the number of rows and siz(2) is the number of columns.

corr_offset = subtract the size of sub template image from xpeak and ypeak which is output of int2sub command.

B) relative offset of position of sub images
rect_offset = rectangular position of template image subtracted from main image

Total offset is as following:

```
offset = corr_offset + rect_offset
xoffset = offset(1)
yoffset = offset(2)
```

Step 4:Figure out where object falls inside of main image

Then round of the offset position by adding one in x and y poison and size of template image.
Example:

```
xbegin = round (xoffset+1)
xend   = round(xoffset+ size(template,2))
ybegin = round(yoffset+1)
yend   = round(yoffset+size(onion,1))
```

Step 5: See if the object Image was extracted from the original Image

Use above xbegin xend , ybegin yend values for extracting matched image from main image as with same size which is in a template.

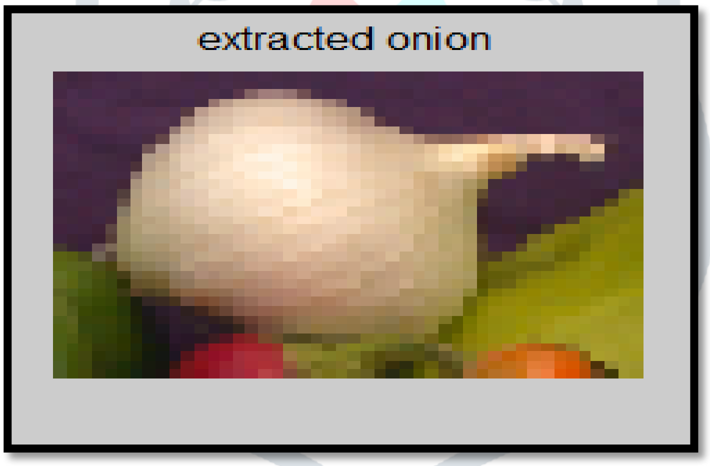


Figure 7 extracted onion

Step 6: check extracted object is matching with original template or not with respect to size and value.

This can be done by using ‘isequal’ function which can compare equality between two images by size and value and second one is again cross correlation function which gives value nearer to 1.00 if two images are match.

If the extracted object is similar to template image, display image which is extracted massage and carried out next steps otherwise display object is not matched.

Till this step we successfully detect the object from original image. Next steps can be done if required for recovering object.

Step 7: Pad the Onion Image to the Size of the Peppers Image

Pad the onion image to overlay on peppers, using the offset determined above. Make other unmatched portion of main image black.

1st blackout complete main image then overlap extracted image one that to get following output

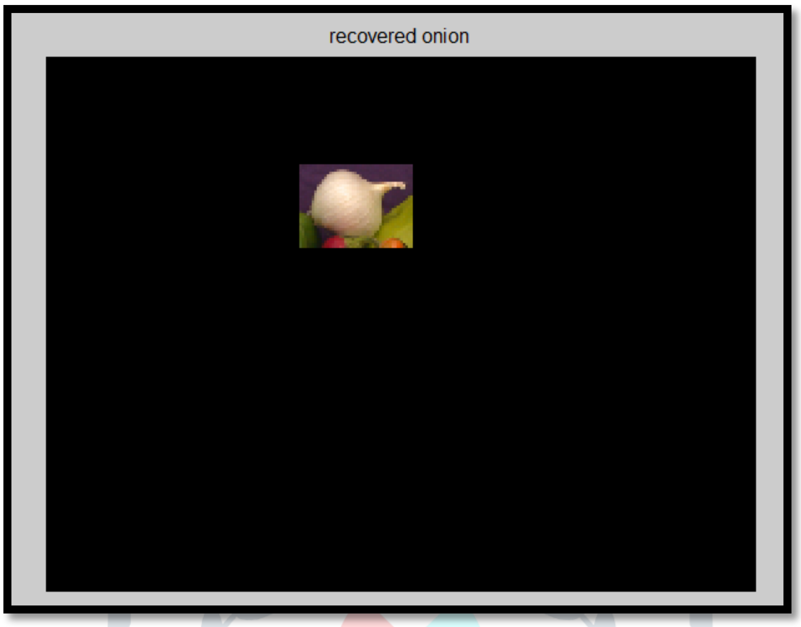
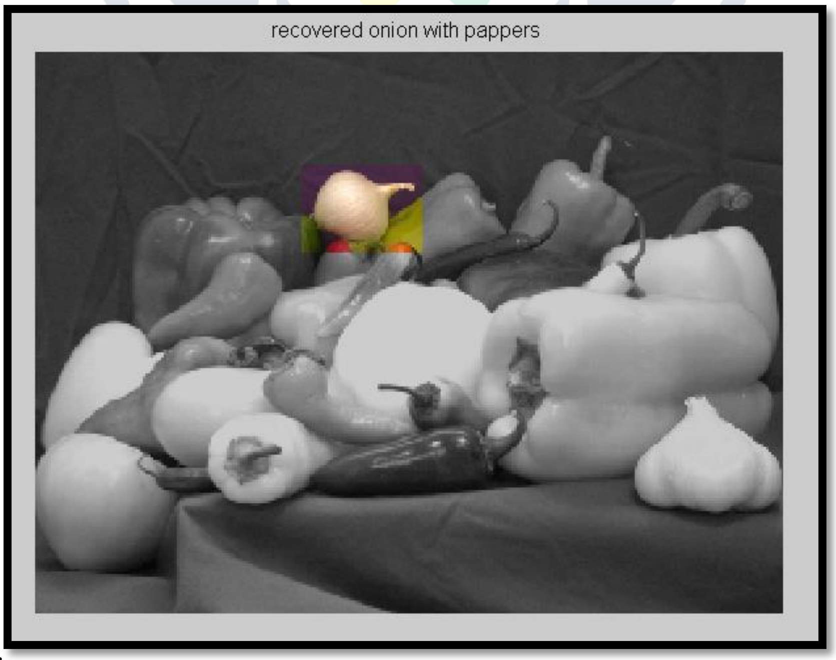


Figure 8 : recovered object

Step 8 Use Alpha Blending to Show Images Together or overlay the image with transparency

Display one plane of the peppers image with the recovered template image using alpha blending or by masking remaining image by different levels of transparency and it pass thought red place to convert in to gray scale.



.

Figure 9 : recovered object with background view

IV. RESULS

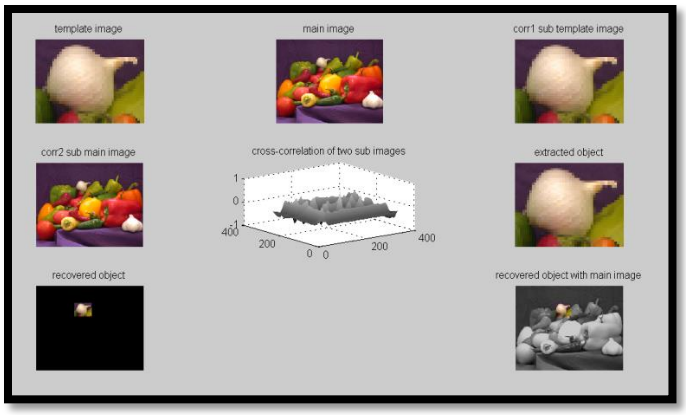


Figure 10: Result 1

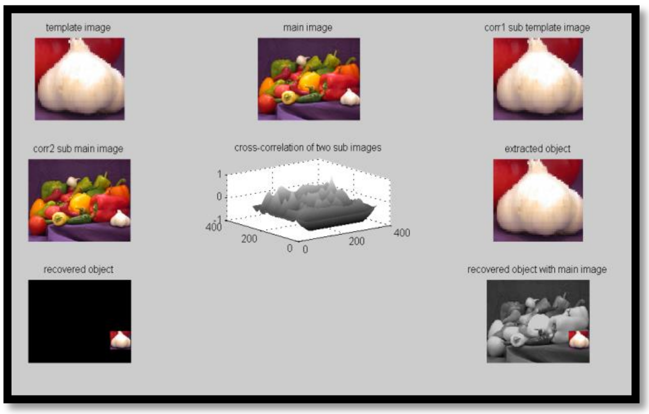


Figure 11: Result2

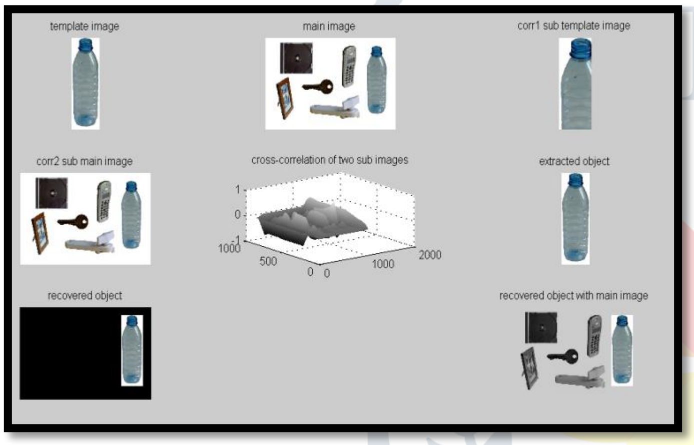


Figure 12: Result 3

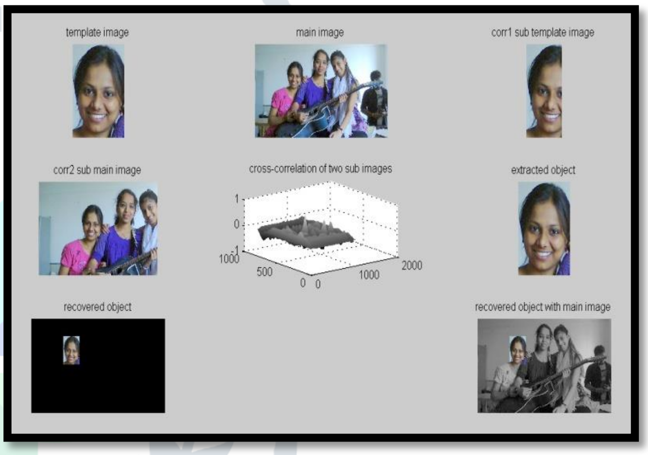


Figure 13: Result3

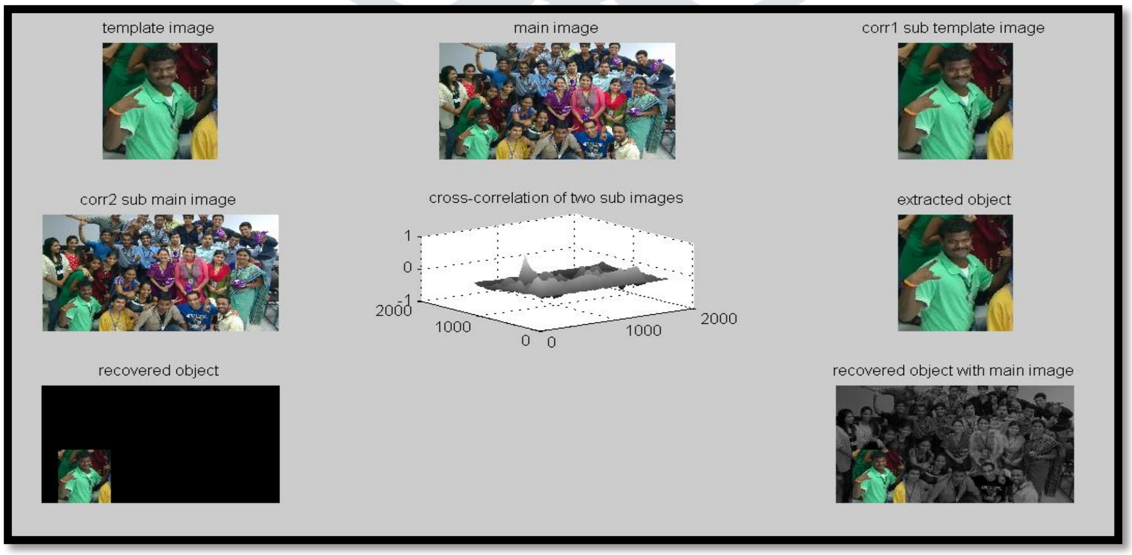


Figure 14: Result 5

V. Conclusion

This paper presented an object detection using template matching and correlation technique. The object detection as well as extraction of detected object successfully achieved by using different images. From this experiment we practically understand the process of correlation, finding out offset from correlation for object detection using given steps with matlab software. We have presented a correlation approach for object detection which is very easily described by steps as included in paper. We conclude that we can use correlation technique for object detection and it is very simple to understand and it doesn’t have any complex algorithm.

VI. References

[1] Sreejith C, Sreeshma K, Suhanna C HA , “Study on Pattern Matching using Eigen Algorithm” Dept. of CSE, MEA, Engg. College, Vengoor, Kerala, India.

[2] Feng Tang and Hai Tao, “Fast Multi-scale Template Matching.Using Binary Features” Department of Computer Engineering, University of California, Santa Cruz, USA.

[3] Hai Tao, Ryan Crabb and Feng Tang, “Non-orthogonal Binary Subspace and its Applications in Computer Vision” Department of Computer Engineering, University of California, Santa Cruz.

[4] Jignesh N Sarvaiya, Suprava Patnaik, Kajal Kothari, “Image Registration Using Log Polar Transform and Phase Correlation to Recover Higher Scale” JOURNAL OF PATTERN DETECTION RESEARCH 7 (2012) 90-105.

[5] Thi Thi Zin, Slmg Shik KOH, and Hiromitsu HAMA, “Novel Template Matching for Extremely Deteriorated Images” Mem. Fac. Eng., Osaka City Univ., Vol. 48, pp. 9-16 (2007).

[6] T. Kawanishi, T. Kurozumi, K. Kashino, S. Takagi, " A Fast Template Matching Algorithm with Adaptive Inner-Subtemplates' Distances" Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Volume:3)

[7] D. Schonfeld, "On the relation of order-statistics filters and template matching: optimal morphological pattern detection," IEEE Trans. on Image Process., vol. 9, no. 5, pp. 945-949, May. (2000).

[8] Mohammmad Gharavi-Alkhansari, "A Fast Globally Optimal Algorithm for Template Matching Using Low-Resolution Pruning"IEEE Trans. Image Processing, vol. 10, no. 4, pp. 526-533, Apr. (2001).

[9] Igor Guskov,"Kernel-based Template Alignment," Proc. a/IEEE Computer Society Conf on Computer Vision and Pattern Detection, pp. 610-617, (2006).

[10] Zhiwei He, Jilin Liu, Hongqing Ma, and Peihong Li, "A new automatic extraction method of container identity codes," IEEE Trans. on Intell. Transp. Syst., vol. 6, no. 1, pp. 72 - 78, Mar. (2005).

[11] R. Etienne-Cwnmings, P. Pouliquen, M. A. Lewis, "Single chip for imaging, color segmentation, histogramming and template matching," Electron Lett., vol. 38, no. 4, pp. 172 - 174, Feb. (2002).

[12] J.P.Lewis, “Fast Template Maching,” Vision Interface, pp. 120- 123, 1995.

[13] Srinivasa Reddy and B. N. Chatterji, “An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration”, IEEE Trans. on Image Processing, Vol. 5, pp.1266-1271, 1996.

[14] Y.Keller, A.Averbuch, M.Israeli, “Pseudo polar-based estimation of large translations, rotations and scalings in images”, IEEE trans. on Image processing, pp.12-22, 2005.